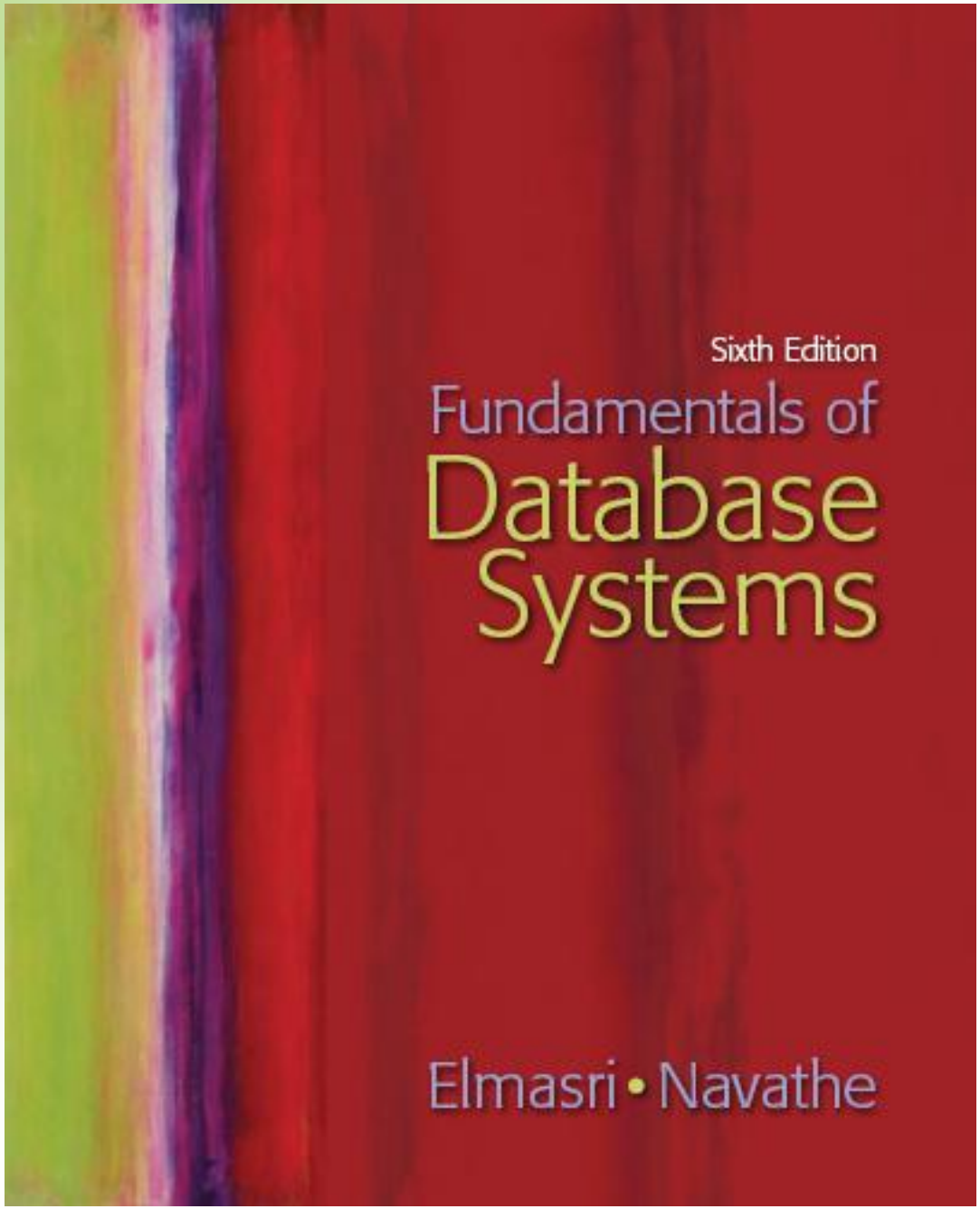
Chapter 29

Overview of Data Warehousing and OLAP



Addison-Wesley
is an imprint of
PEARSON

Purpose of Data Warehousing

- Traditional databases are not optimized for data access only they have to balance the requirement of data access with the need to ensure integrity of data.
- Most of the times the data warehouse users need only read access but, need the access to be fast over a large volume of data.
- Most of the data required for **data warehouse** analysis comes from multiple databases and these analysis are recurrent and predictable to be able to design specific software to meet the requirements.
- There is a great need for tools that provide decision makers with information to make decisions quickly and reliably based on historical data.
- The above functionality is achieved by Data Warehousing and **Online analytical processing (OLAP)**

Introduction, Definitions, and Terminology

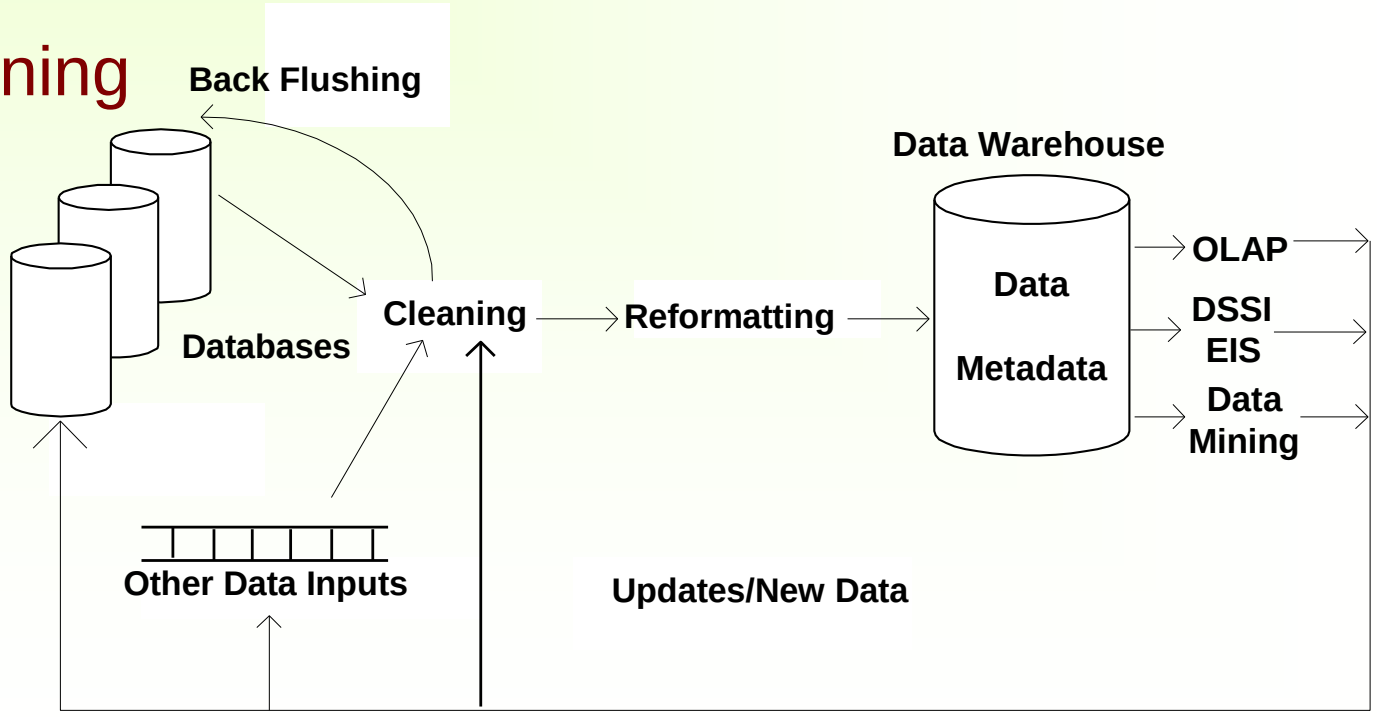
- W. H Inmon characterized a data warehouse as:
  - **"A subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management's decisions."**

Introduction, Definitions, and Terminology

- Data warehouses have the distinguishing characteristic that they are mainly intended for decision support applications.
  - Traditional databases are transactional.
- Applications that data warehouse supports are:
  - **OLAP** (Online Analytical Processing) is a term used to describe the analysis of complex data from the data warehouse.
  - **DSS** (Decision Support Systems) also known as EIS (Executive Information Systems) supports organization's leading decision makers for making complex and important decisions.
  - **Data Mining** is used for knowledge discovery, the process of searching data for unanticipated new knowledge.

Conceptual Structure of Data Warehouse

- Data Warehouse processing involves
  - Cleaning and reformatting of data
  - OLAP
  - Data Mining



Comparison with Traditional Databases

- Data Warehouses are mainly optimized for appropriate data access.
  - Traditional databases are transactional and are optimized for both access mechanisms and integrity assurance measures.
- Data warehouses emphasize more on historical data as their main purpose is to support time-series and trend analysis.
- Compared with transactional databases, data warehouses are nonvolatile.
- In transactional databases transaction is the mechanism change to the database. By contrast information in data warehouse is relatively coarse grained and refresh policy is carefully chosen, usually incremental.

Characteristics of Data Warehouses

- Multidimensional conceptual view
- Generic dimensionality
- Unlimited dimensions and aggregation levels
- Unrestricted cross-dimensional operations
- Dynamic sparse matrix handling
- Client-server architecture
- Multi-user support
- Accessibility
- Transparency
- Intuitive data manipulation
- Consistent reporting performance
- Flexible reporting

Classification of Data Warehouses

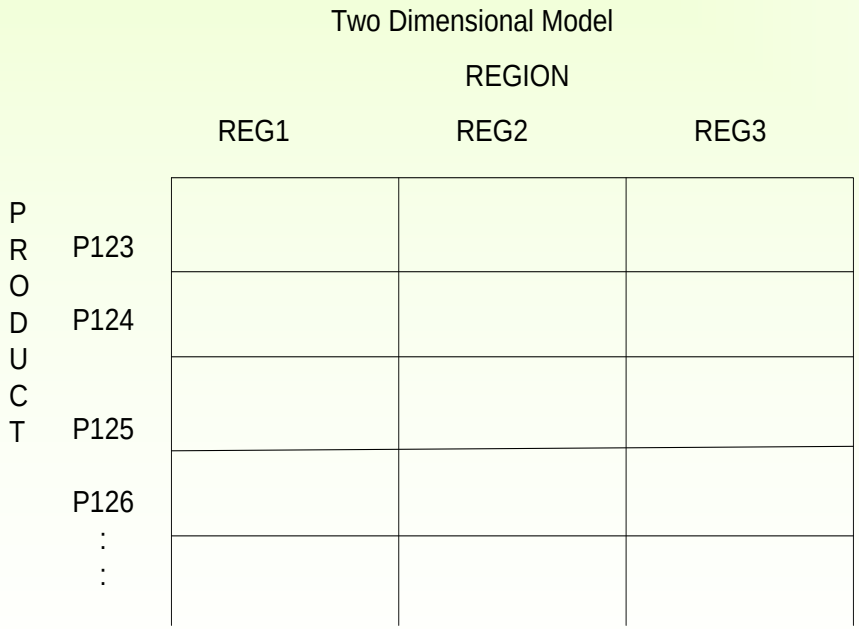
- Generally, Data Warehouses are an order of magnitude larger than the source databases.
- The sheer volume of data is an issue, based on which Data Warehouses could be classified as follows.
  - Enterprise-wide data warehouses
    - They are huge projects requiring massive investment of time and resources.
  - Virtual data warehouses
    - They provide views of operational databases that are materialized for efficient access.
  - Data marts
    - These are generally targeted to a subset of organization, such as a department, and are more tightly focused.

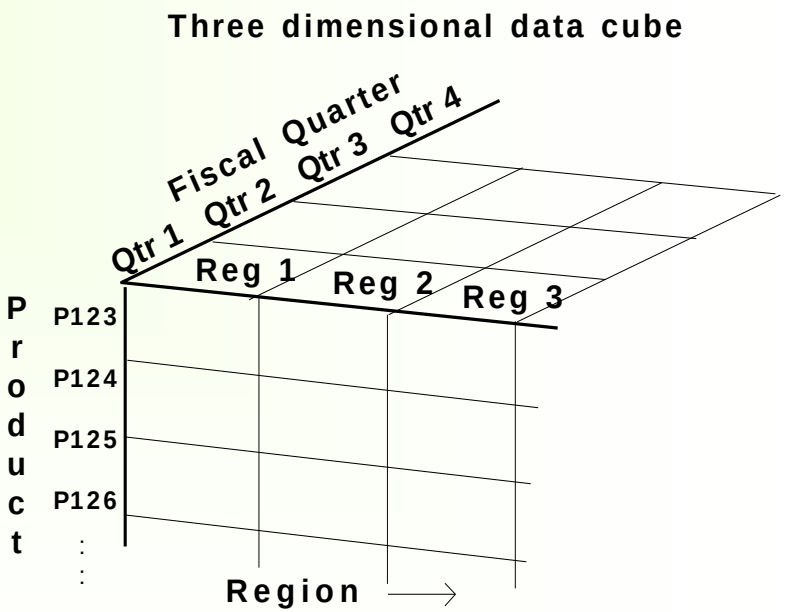
Data Modeling for Data Warehouses

- Traditional Databases generally deal with two-dimensional data (similar to a spread sheet).
  - However, querying performance in a multi-dimensional data storage model is much more efficient.
- Data warehouses can take advantage of this feature as generally these are
  - Non volatile
  - The degree of predictability of the analysis that will be performed on them is high.

Data Modeling for Data Warehouses

- Example of Two- Dimensional vs. Multi-Dimensional





Data Modeling for Data Warehouses

- Advantages of a multi-dimensional model
  - Multi-dimensional models lend themselves readily to hierarchical views in what is known as roll-up display and drill-down display.
  - The data can be directly queried in any combination of dimensions, bypassing complex database queries.

Multi-dimensional Schemas

- Multi-dimensional schemas are specified using:
  - **Dimension table**
    - It consists of tuples of attributes of the dimension.
  - **Fact table**
    - Each tuple is a recorded fact. This fact contains some measured or observed variable (s) and identifies it with pointers to dimension tables. The fact table contains the data, and the dimensions to identify each tuple in the data.

Multi-dimensional Schemas

- Two common multi-dimensional schemas are
  - **Star schema:**
    - Consists of a fact table with a single table for each dimension
  - **Snowflake Schema:**
    - It is a variation of star schema, in which the dimensional tables from a star schema are organized into a hierarchy by normalizing them.

Multi-dimensional Schemas

- **Star schema**:
  - Consists of a fact table with a single table for each dimension.

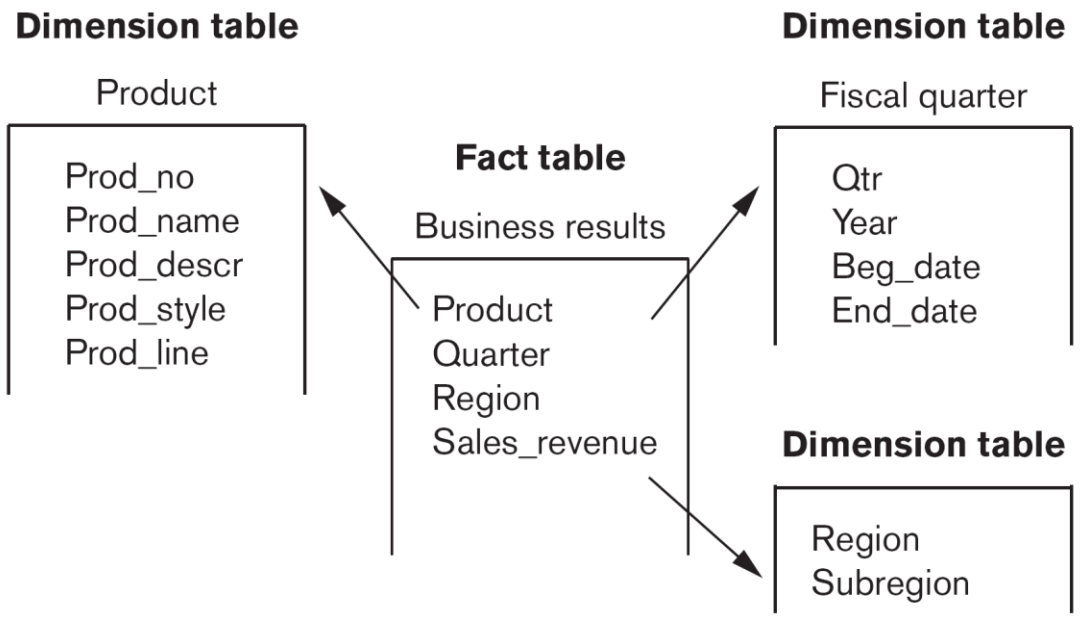
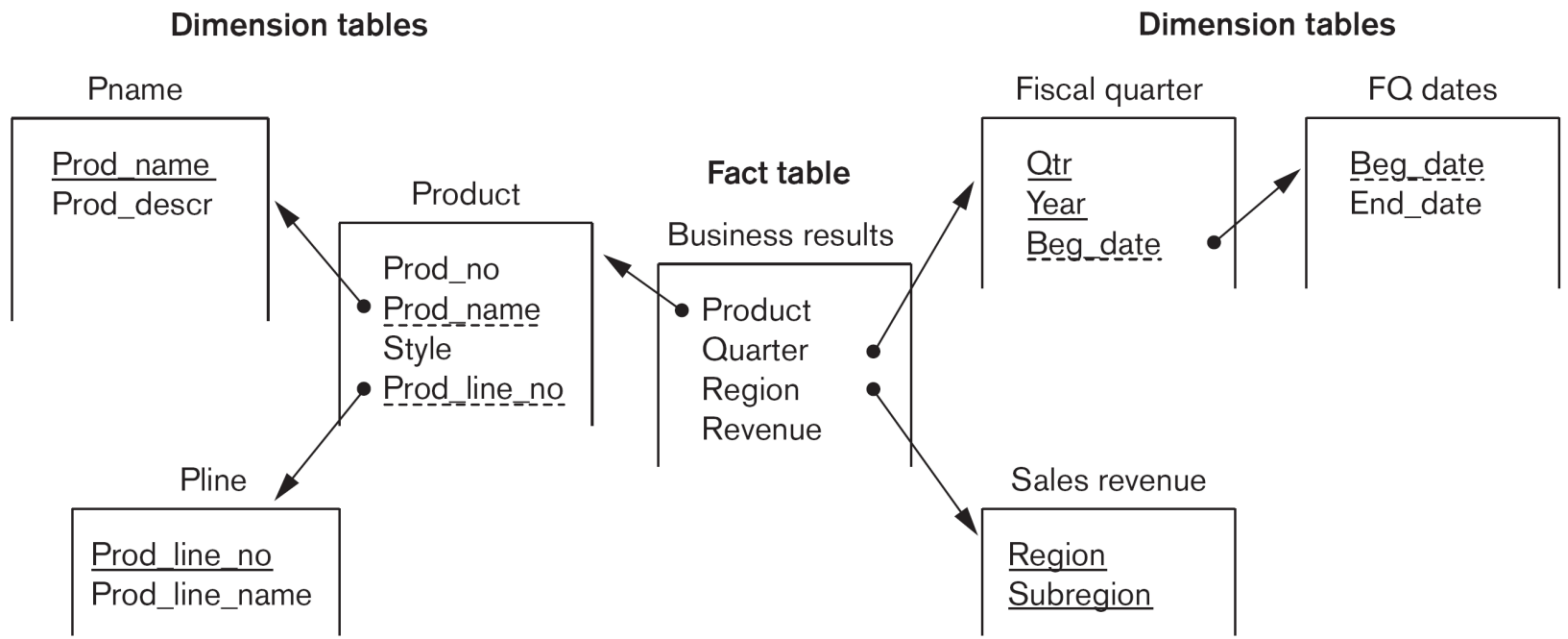


Figure 29.7
A star schema with fact and dimensional tables.

Multi-dimensional Schemas

- **Snowflake Schema:**
  - It is a variation of star schema, in which the dimensional tables from a star schema are organized into a hierarchy by normalizing them.

Figure 29.8
A snowflake schema.



Multi-dimensional Schemas

- **Fact Constellation**
  - Fact constellation is a set of tables that share some dimension tables. However, fact constellations limit the possible queries for the warehouse.

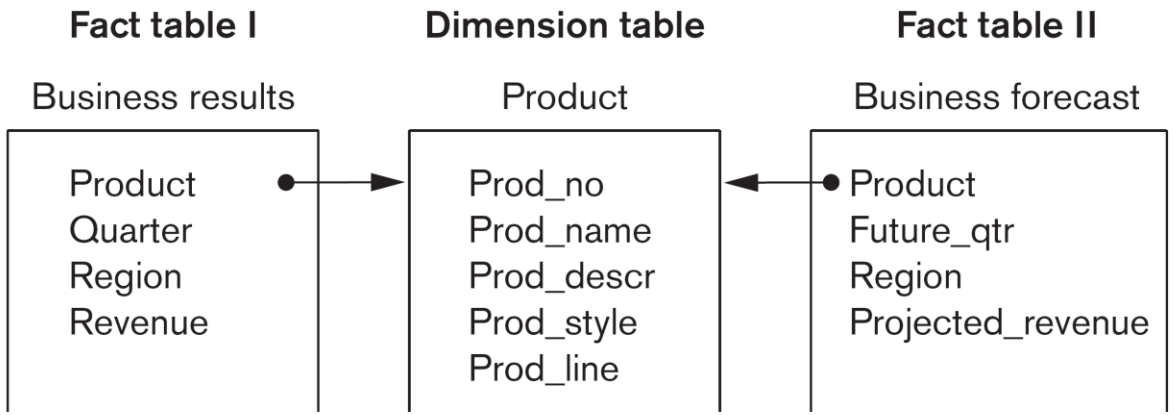


Figure 29.9
A fact constellation.

Multi-dimensional Schemas

- **Indexing**
  - Data warehouse also utilizes indexing to support high performance access.
  - A technique called bitmap indexing constructs a bit vector for each value in domain being indexed.
  - Indexing works very well for domains of low cardinality.

Building A Data Warehouse

- The builders of Data warehouse should take a broad view of the anticipated use of the warehouse.
  - The design should support ad-hoc querying
  - An appropriate schema should be chosen that reflects the anticipated usage.

Building A Data Warehouse

- The Design of a Data Warehouse involves following steps.
  - Acquisition of data for the warehouse.
  - Ensuring that Data Storage meets the query requirements efficiently.
  - Giving full consideration to the environment in which the data warehouse resides.

Building A Data Warehouse

- Acquisition of data for the warehouse
  - The data must be extracted from multiple, heterogeneous sources.
  - Data must be formatted for consistency within the warehouse.
  - The data must be cleaned to ensure validity.
    - Difficult to automate cleaning process.
    - Back flushing, upgrading the data with cleaned data.

Building A Data Warehouse

- Acquisition of data for the warehouse (contd.)
  - The data must be fitted into the data model of the warehouse.
  - The data must be loaded into the warehouse.
    - Proper design for refresh policy should be considered.

Building A Data Warehouse

- Storing the data according to the data model of the warehouse
- Creating and maintaining required data structures
- Creating and maintaining appropriate access paths
- Providing for time-variant data as new data are added
- Supporting the updating of warehouse data.
- Refreshing the data
- Purging data

Building A Data Warehouse

- Usage projections
- The fit of the data model
- Characteristics of available resources
- Design of the metadata component
- Modular component design
- Design for manageability and change
- Considerations of distributed and parallel architecture
  - Distributed vs. federated warehouses

Functionality of a Data Warehouse

- Functionality that can be expected:
  - **Roll-up**: Data is summarized with increasing generalization
  - **Drill-Down**: Increasing levels of detail are revealed
  - **Pivot**: Cross tabulation is performed
  - **Slice** and dice: Performing projection operations on the dimensions.
  - **Sorting**: Data is sorted by ordinal value.
  - **Selection**: Data is available by value or range.
  - **Derived attributes**: Attributes are computed by operations on stored derived values.

Warehouse vs. Data Views

- Views and data warehouses are alike in that they both have read-only extracts from the databases.
- However, data warehouses are different from views in the following ways:
  - Data Warehouses exist as persistent storage instead of being materialized on demand.
  - Data Warehouses are not usually relational, but rather multi-dimensional.
  - Data Warehouses can be indexed for optimization.
  - Data Warehouses provide specific support of functionality.
  - Data Warehouses deals huge volumes of data that is contained generally in more than one database.

Difficulties of implementing Data Warehouses

- Lead time is huge in building a data warehouse
  - Potentially it takes years to build and efficiently maintain a data warehouse.
- Both quality and consistency of data are major concerns.
- Revising the usage projections regularly to meet the current requirements.
  - The data warehouse should be designed to accommodate addition and attrition of data sources without major redesign
- Administration of data warehouse would require far broader skills than are needed for a traditional database.

Open Issues in Data Warehousing

- Data cleaning, indexing, partitioning, and views could be given new attention with perspective to data warehousing.
- Automation of
  - data acquisition
  - data quality management
  - selection and construction of access paths and structures
  - self-maintainability
  - functionality and performance optimization
- Incorporating of domain and business rules appropriately into the warehouse creation and maintenance process more intelligently.

Recap

- Purpose of Data Warehousing
- Introduction, Definitions, and Terminology
- Comparison with Traditional Databases
- Characteristics of data Warehouses
- Classification of Data Warehouses
- Multi-dimensional Schemas
- Building A Data Warehouse
- Functionality of a Data Warehouse
- Warehouse vs. Data Views
- Implementation difficulties and open issues